



Coutts, Fraser K. and Thompson, Keith and Proudler, Ian K. and Weiss, Stephan (2017) Restricted update sequential matrix diagonalisation for parahermitian matrices. In: IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2017-12-10 - 2017-12-13. (In Press) ,

This version is available at <https://strathprints.strath.ac.uk/61930/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<https://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

The Strathprints institutional repository (<https://strathprints.strath.ac.uk>) is a digital archive of University of Strathclyde research outputs. It has been developed to disseminate open access research outputs, expose data about those outputs, and enable the management and persistent access to Strathclyde's intellectual output.

Restricted Update Sequential Matrix Diagonalisation for Parahermitian Matrices

Fraser K. Coutts*, Keith Thompson*, Ian K. Proudler*,[†], Stephan Weiss*

* Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland

[†] School of Electrical, Electronics & Systems Engineering, Loughborough Univ., Loughborough, UK
 {fraser.coutts, keith.thompson, ian.proudler, stephan.weiss}@strath.ac.uk

Abstract—A number of algorithms capable of iteratively calculating a polynomial matrix eigenvalue decomposition (PEVD) have been introduced. The PEVD is an extension of the ordinary EVD to polynomial matrices and will diagonalise a parahermitian matrix using paraunitary operations. This paper introduces a novel restricted update approach for the sequential matrix diagonalisation (SMD) PEVD algorithm, which can be implemented with minimal impact on algorithm accuracy and convergence. We demonstrate that by using the proposed restricted update SMD (RU-SMD) algorithm instead of SMD, PEVD complexity and execution time can be significantly reduced. This reduction impacts on a number of broadband multichannel problems.

I. INTRODUCTION

Polynomial matrix representations can be used to express broadband multichannel problems [1]. Such formulations can be used in a number of areas, including broadband MIMO precoding and equalisation [2], polyphase analysis and synthesis matrices for filter banks [3], broadband beamforming [4], [5], and broadband angle of arrival estimation [6], [7]. Typically, these problems involve parahermitian polynomial matrices, which are identical to their parahermitian conjugate, i.e., $\mathbf{R}(z) = \tilde{\mathbf{R}}(z) = \mathbf{R}^H(1/z^*)$ [3]. This matrix $\mathbf{R}(z)$ can arise as the z -transform of a space-time covariance matrix $\mathbf{R}[\tau]$.

As an extension of the eigenvalue decomposition to parahermitian matrices, a polynomial matrix eigenvalue decomposition (PEVD) has been defined in [8]. The PEVD uses finite impulse response (FIR) paraunitary matrices [9] to approximately diagonalise and spectrally majorise [10] $\mathbf{R}(z)$.

Existing PEVD algorithms include sequential matrix diagonalisation (SMD) [11], second-order sequential best rotation (SBR2) [8], and various evolutions of the algorithm families [12]–[15]. Each of these algorithms use an iterative approach to approximately diagonalise a parahermitian matrix. For matrices of high dimensionality, these algorithms can be computationally costly to compute; therefore, any cost savings will be advantageous for applications.

In an effort to reduce the cost of PEVD algorithms, previous work in [8], [16]–[19] has focussed on the trimming of polynomial matrices to curb growth in order. Techniques in [20], [21] have successfully reduced the complexity of existing PEVD algorithms through the removal of algorithmic redundancy. Recent work in [22], [23] describes a low-complexity divide-and-conquer approach for the PEVD.

Research in [15], [21] has shown that restricting the search space of iterative PEVD algorithms to a subset of lags around lag zero of a parahermitian matrix can bring performance

gains with little impact on algorithm convergence. However, the entire parahermitian matrix must still be updated at each iteration in these approaches. This paper expands upon this idea by introducing a novel restricted update SMD (RU-SMD) algorithm which not only restricts the search space of the SMD algorithm, but also restricts the portion of the parahermitian matrix that is updated at each iteration. The update step of SMD is its most computationally costly operation [11]; thus, a reduction in the complexity of this step is useful.

Below, Sec. II will provide a brief overview over the SMD algorithm, which can be modified to accommodate the restricted update procedure outlined in Sec. III. Simulation results comparing the performance of the standard and proposed approaches are presented in Sec. IV, with conclusions drawn in Sec. V.

II. SEQUENTIAL MATRIX DIAGONALISATION

This section reviews the SMD algorithm [11] in Sec. II-A, with an assessment of the algorithmic cost in Sec. II-B.

A. Algorithm Overview

The SMD algorithm approximates the PEVD using a series of elementary paraunitary operations to iteratively diagonalise a parahermitian matrix $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$ and its associated coefficient matrix, $\mathbf{R}[\tau]$.

Upon initialisation, the algorithm diagonalises the lag-zero coefficient matrix $\mathbf{R}[0]$ by means of its modal matrix $\mathbf{Q}^{(0)}$; i.e., $\mathbf{S}^{(0)}(z) = \mathbf{Q}^{(0)} \mathbf{R}(z) \mathbf{Q}^{(0)H}$. The unitary $\mathbf{Q}^{(0)}$ — obtained from the EVD of the lag-zero slice $\mathbf{R}[0]$ — is applied to all coefficient matrices $\mathbf{R}[\tau] \forall \tau$, and initialises $\mathbf{H}^{(0)}(z) = \mathbf{Q}^{(0)}$.

In the i th step, $i = 1, 2, \dots, I$, the SMD algorithm computes

$$\begin{aligned} \mathbf{S}^{(i)}(z) &= \mathbf{U}^{(i)}(z) \mathbf{S}^{(i-1)}(z) \tilde{\mathbf{U}}^{(i)}(z) \\ \mathbf{H}^{(i)}(z) &= \mathbf{U}^{(i)}(z) \mathbf{H}^{(i-1)}(z) \end{aligned} \quad , \quad (1)$$

in which

$$\mathbf{U}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{\Lambda}^{(i)}(z) . \quad (2)$$

The product in (2) consists of a paraunitary delay matrix

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\left\{ \underbrace{1 \dots 1}_{k^{(i)}-1} z^{-\tau^{(i)}} \underbrace{1 \dots 1}_{M-k^{(i)}} \right\} , \quad (3)$$

and a unitary matrix $\mathbf{Q}^{(i)}$, with the result that $\mathbf{U}^{(i)}(z)$ in (2) is paraunitary. For subsequent discussion, it is convenient to define intermediate variables $\mathbf{S}^{(i)'}(z)$ and $\mathbf{H}^{(i)'}(z)$ where

$$\begin{aligned} \mathbf{S}^{(i)'}(z) &= \mathbf{\Lambda}^{(i)}(z) \mathbf{S}^{(i-1)}(z) \tilde{\mathbf{\Lambda}}^{(i)}(z) \\ \mathbf{H}^{(i)'}(z) &= \mathbf{\Lambda}^{(i)}(z) \mathbf{H}^{(i-1)}(z) \end{aligned} \quad , \quad (4)$$

and

$$\begin{aligned} \mathbf{S}^{(i)}(z) &= \mathbf{Q}^{(i)} \mathbf{S}^{(i)'}(z) \mathbf{Q}^{(i)H} \\ \mathbf{H}^{(i)}(z) &= \mathbf{Q}^{(i)} \mathbf{H}^{(i)'}(z) \end{aligned} \quad (5)$$

Matrices $\mathbf{A}^{(i)}(z)$ and $\mathbf{Q}^{(i)}$ are selected based on the position of the dominant off-diagonal column in $\mathbf{S}^{(i-1)}(z)$ — $\mathbf{S}^{(i-1)}[\tau]$, as identified by the parameter set

$$\{k^{(i)}, \tau^{(i)}\} = \arg \max_{k, \tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2, \quad (6)$$

where

$$\|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 = \sqrt{\sum_{m=1, m \neq k}^M |\mathbf{s}_{m,k}^{(i-1)}[\tau]|^2} \quad (7)$$

and $\mathbf{s}_{m,k}^{(i-1)}[\tau]$ represents the element in the m th row and k th column of the coefficient matrix at lag τ , $\mathbf{S}^{(i-1)}[\tau]$.

The shifting process in (4) moves the dominant off-diagonal row and column into the zero lag coefficient matrix $\mathbf{S}^{(i)'}[0]$. The off-diagonal energy in the shifted row and column is then transferred onto the diagonal by the unitary matrix $\mathbf{Q}^{(i)}$ in (5), which diagonalises $\mathbf{S}^{(i)'}[0]$ by means of an ordered EVD.

Iterations continue for I steps until $\mathbf{S}^{(I)}(z)$ is sufficiently diagonalised with dominant off-diagonal column norm

$$\max_{k, \tau} \|\hat{\mathbf{s}}_k^{(I)}[\tau]\|_2 \leq \epsilon, \quad (8)$$

where the value of ϵ is chosen to be arbitrarily small. On completion, SMD generates an approximate PEVD given by

$$\mathbf{D}(z) = \mathbf{S}^{(I)}(z) = \mathbf{F}(z) \mathbf{R}(z) \tilde{\mathbf{F}}(z), \quad (9)$$

where $\mathbf{F}(z)$ is a concatenation of the paraunitary matrices:

$$\mathbf{F}(z) = \mathbf{H}^{(I)}(z) = \mathbf{U}^{(I)}(z) \cdots \mathbf{U}^{(0)}(z) = \prod_{i=0}^I \mathbf{U}^{(I-i)}(z).$$

Truncation of outer coefficients of $\mathbf{H}^{(i)}(z)$ with small Frobenius norm $\|\cdot\|_F$ is used to limit growth in order, whereby the maximum and minimum lags of $\mathbf{H}^{(i)}(z)$ at iteration i are reduced from τ_1 and τ_2 to $\tilde{\tau}_1$ and $\tilde{\tau}_2$, respectively, such that

$$\sum_{\tau=\tilde{\tau}_1+1}^{\tau_1} \|\mathbf{H}^{(i)}[\tau]\|_F^2 < \frac{\mu \sum_{\tau} \|\mathbf{H}^{(i)}[\tau]\|_F^2}{2} > \sum_{\tau=\tau_2}^{\tilde{\tau}_2-1} \|\mathbf{H}^{(i)}[\tau]\|_F^2. \quad (10)$$

Truncation of $\mathbf{S}^{(i)}(z)$ is similar, with its maximum and minimum lags reduced from τ_3 and $-\tau_3$ to $\tilde{\tau}_3$ and $-\tilde{\tau}_3$, such that

$$\sum_{\tau=\tilde{\tau}_3+1}^{\tau_3} \|\mathbf{S}^{(i)}[\tau]\|_F^2 < \frac{\mu \sum_{\tau} \|\mathbf{S}^{(i)}[\tau]\|_F^2}{2}. \quad (11)$$

B. Algorithm Complexity

At the i th iteration, the length of $\mathbf{S}^{(i)'}(z)$ is equal to $L\{\mathbf{S}^{(i)'}\}$, where $L\{\cdot\}$ computes the length of a polynomial matrix. For (5), every matrix-valued coefficient in $\mathbf{S}^{(i)'}(z)$ must be left- and right-multiplied with a unitary matrix. Accounting for a multiplication of $2 M \times M$ matrices by M^3 MACs, a total of $2L\{\mathbf{S}^{(i)'}\}M^3$ MACs arise to generate $\mathbf{S}^{(i)}(z)$. Every matrix-valued coefficient in $\mathbf{H}^{(i)'}(z)$ must also be left-multiplied with a unitary matrix; thus, a total of $L\{\mathbf{H}^{(i)'}\}M^3$ MACs arise to generate $\mathbf{H}^{(i)'}(z)$.

The cumulative complexity of the SMD algorithm over I iterations can therefore be approximated as

$$C_{\text{SMD}}(I) = M^3 \sum_{i=0}^I (2L\{\mathbf{S}^{(i)'}\} + L\{\mathbf{H}^{(i)'}\}). \quad (12)$$

III. RESTRICTED UPDATE SEQUENTIAL MATRIX DIAGONALISATION

Research in [15], [21] has shown that restricting the search space of iterative PEVD algorithms to a subset of lags around lag zero of a parahermitian matrix can bring performance gains with little impact on algorithm convergence. This section expands upon this idea by introducing a novel restricted update SMD (RU-SMD) algorithm which not only restricts the search space of the SMD algorithm, but also restricts the portion of the parahermitian matrix that is updated at each iteration.

An overview of the method is given in Sec. III-A, with more detailed descriptions of the major components in Sec. III-B and Sec. III-C. Finally, Sec. III-D describes the algorithmic complexity of the approach.

A. Overview

Similarly to SMD, the RU-SMD algorithm approximates a PEVD by iteratively diagonalising a parahermitian matrix $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$ over $i = 0 \dots I$ iteration steps. Over the course of these iterations, the search space contracts piecewise strictly monotonically. This restriction limits the number of search operations, but also reduces the computations required to update the increasingly diagonalised parahermitian matrix. The search space contracts until order zero is reached and the search window only includes the zero lag matrix. After this, in a so-called regeneration step, the search window is maximised and thereafter again contracts monotonically over the following iterations. The maximum β of index $\alpha = 0 \dots \beta$, which counts the number of regenerations, is not known a priori.

Following the α th regeneration step, in the i th iteration of RU-SMD, $\mathbf{S}^{(i)}(z) = \mathbf{R}_{(\alpha)}(z)$. Note, $\mathbf{R}_{(0)}(z) = \mathbf{Q}^{(0)} \mathbf{R}(z) \mathbf{Q}^{(0)H}$, where $\mathbf{Q}^{(0)}$ diagonalises coefficient matrix $\mathbf{R}[0]$. The restricted update stage of RU-SMD restricts the search and update steps of the SMD algorithm to only consider an iteratively decreasing selection of lags of $\mathbf{S}^{(i)}(z)$ around lag zero at the i th iteration. When the search space of $\mathbf{S}^{(i)}(z)$ reaches zero, the restricted update stage has produced a paraunitary matrix $\mathbf{F}_{(\alpha)}(z)$ such that matrix $\mathbf{R}_{(\alpha+1)}(z) = \mathbf{F}_{(\alpha)}(z) \mathbf{R}_{(\alpha)}(z) \tilde{\mathbf{F}}_{(\alpha)}(z)$ — which is generated during the matrix regeneration stage — is more diagonal than $\mathbf{R}_{(\alpha)}(z)$.

Algorithm 1 gives the pseudocode for RU-SMD. Output matrices $\mathbf{F}(z)$ and $\mathbf{D}(z)$ contain polynomial eigenvectors and eigenvalues, respectively. More detail of the algorithm's operation is provided in Sec. III-B and Sec. III-C. Convergence of the SMD algorithm has already been proven [11]; the proof also holds for this RU-SMD algorithm.

B. Restricted Update

The restricted update step of RU-SMD functions similarly to the update step of SMD in Sec. II-A; however, the key difference is that RU-SMD increasingly restricts the number of lags of $\mathbf{S}^{(i)}(z)$ that are updated at each iteration i . Algorithm 2 provides pseudocode for the *Restricted_update* function, whose operation is discussed below.

From a parahermitian matrix $\mathbf{R}_{(\alpha)}(z) \in \mathbb{C}^{M \times M}$ input to the *Restricted_update* function for index α , during the i th iteration of RU-SMD, a matrix $\mathbf{S}^{(i-1)}(z) = \mathbf{R}_{(\alpha)}(z)$ with maximum lag $\tau_{\max}^{(i)}$ is formed. As in the standard SMD

Input: $R(z)$, μ , ϵ , I
Output: $D(z)$, $F(z)$
Find eigenvectors $\mathbf{Q}^{(0)}$ that diagonalise $\mathbf{R}[0]$
 $\mathbf{R}_{(0)}(z) = \mathbf{Q}^{(0)} \mathbf{R}(z) \mathbf{Q}^{(0)H}$, $\mathbf{F}'_{(0)}(z) = \mathbf{Q}^{(0)}$, $\alpha = 0$, $i = 0$,
stop = 0
while stop = 0 **do**
 $[\mathbf{F}_{(\alpha)}(z), i, \text{stop}] = \text{Restricted_update}(\mathbf{R}_{(\alpha)}(z), \mu, \epsilon, I, i)$
 Regenerate matrix:
 $\mathbf{F}'_{(\alpha+1)}(z) = \mathbf{F}_{(\alpha)}(z) \mathbf{F}'_{(\alpha)}(z)$
 $\mathbf{R}_{(\alpha+1)}(z) = \mathbf{F}_{(\alpha)}(z) \mathbf{R}_{(\alpha)}(z) \tilde{\mathbf{F}}_{(\alpha)}(z)$
 Truncate $\mathbf{F}'_{(\alpha+1)}(z)$, $\mathbf{R}_{(\alpha+1)}(z)$ according to (10), (11)
 $\alpha = \alpha + 1$
end
 $\mathbf{F}(z) = \mathbf{F}'_{(\alpha)}(z)$, $\mathbf{D}(z) = \mathbf{R}_{(\alpha)}(z)$

Algorithm 1: RU-SMD Algorithm

Input: $\mathbf{R}_{(\alpha)}(z)$, μ , ϵ , I , i
Output: $\mathbf{F}_{(\alpha)}(z)$, i , stop
 $\mathbf{S}^{(i)}(z) = \mathbf{R}_{(\alpha)}(z)$, $\mathbf{H}^{(i)}(z) = \mathbf{I}_{M \times M}$, stop = 0
do
 $i = i + 1$
 Find $\{k^{(i)}, \tau^{(i)}\}$ from (6); generate $\mathbf{\Lambda}^{(i)}(z)$ from (3)
 $\mathbf{S}^{(i)'}(z) = \mathbf{\Lambda}^{(i)}(z) \mathbf{S}^{(i-1)}(z) \tilde{\mathbf{\Lambda}}^{(i)}(z)$
 Find eigenvectors $\mathbf{Q}^{(i)}$ that diagonalise $\mathbf{S}^{(i)'}[0]$
 $\tau_{\max}^{(i)}$ is maximum lag of $\mathbf{S}^{(i-1)}(z)$
 $\mathbf{S}^{(i)''}(z) = \sum_{\tau=-\tau_{\max}^{(i)}+|\tau^{(i)}|}^{\tau_{\max}^{(i)}-|\tau^{(i)}|} \mathbf{S}^{(i)'}[\tau] z^{-\tau}$
 $\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{S}^{(i)''}(z) \mathbf{Q}^{(i)H}$
 $\mathbf{H}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{H}^{(i-1)}(z) = \mathbf{Q}^{(i)} \mathbf{\Lambda}^{(i)}(z) \mathbf{H}^{(i-1)}(z)$
 Truncate $\mathbf{H}^{(i)}(z)$ according to (10)
 if $i > I$ or (8) satisfied **then**
 | stop = 1;
 end
while stop = 0 and $(\tau_{\max}^{(i)} - |\tau^{(i)}|) > 0$
 $\mathbf{F}_{(\alpha)}(z) = \mathbf{H}^{(i)}(z)$

Algorithm 2: Restricted_update Function

algorithm, the $k^{(i)}$ th column and row with maximum energy are found and shifted by $\tau^{(i)}$ to the zero lag using a delay matrix $\mathbf{\Lambda}^{(i)}(z)$ to produce $\mathbf{S}^{(i)'}(z)$.

A unitary matrix $\mathbf{Q}^{(i)}$ is generated from an EVD of the zero lag $\mathbf{S}^{(i)'}[0]$, but is only applied to update region matrix $\mathbf{S}^{(i)''}(z)$, which contains the central $(2(\tau_{\max}^{(i)} - |\tau^{(i)}|) + 1)$ lags of $\mathbf{S}^{(i)'}(z)$. Thus, matrix $\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{S}^{(i)''}(z) \mathbf{Q}^{(i)H}$ is formed, which has maximum lag $\tau_{\max}^{(i+1)} = \tau_{\max}^{(i)} - |\tau^{(i)}|$.

The coefficients of $\mathbf{S}^{(i)}[\tau]$ at lags $|\tau| > \tau_{\max}^{(i+1)}$, which are zero by definition — and not obtained from the transformation of $\mathbf{S}^{(i-1)}(z)$ — must be kept outside of the update region in the next iteration, $\mathbf{S}^{(i+1)''}(z)$, if the accuracy of the decomposition is to be maintained. To guarantee that these coefficients are excluded, the update region must shrink by the maximum possible distance that the coefficients can travel towards the zero lag, $|\tau^{(i+1)}|$. That is, $\mathbf{S}^{(i+1)''}(z)$ should only contain the central $2(\tau_{\max}^{(i+1)} - |\tau^{(i+1)}|) + 1$ lags of $\mathbf{S}^{(i+1)}(z)$.

Iterations of this process continue in the same manner until then end of some iteration $I_{(\alpha)}$, when the maximum lag

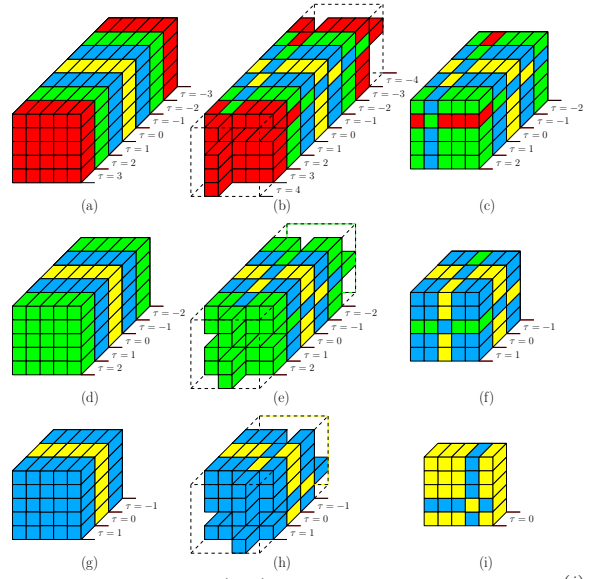


Fig. 1. (a) Original matrix $\mathbf{S}^{(i-1)}(z) \in \mathbb{C}^{5 \times 5}$ with maximum lag $\tau_{\max}^{(i)} = 3$ is input to *Restricted_update*; (b) shifting of row and column energy to zero lag ($k^{(i)} = 2$, $\tau^{(i)} = 1$); (c) central matrix with maximum lag $(\tau_{\max}^{(i)} - |\tau^{(i)}|) = 2$, $\mathbf{S}^{(i)''}(z)$, is extracted. (d) $\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{S}^{(i)''}(z) \mathbf{Q}^{(i)H}$; (e) $k^{(i+1)} = 3$, $\tau^{(i+1)} = 1$; (f) $\mathbf{S}^{(i+1)''}(z)$ extracted. (g) $\mathbf{S}^{(i+1)}(z)$; (h) $k^{(i+2)} = 4$, $\tau^{(i+2)} = 1$; (i) $\mathbf{S}^{(i+2)''}(z)$ is extracted.

of matrix $\mathbf{S}^{(I_{(\alpha)})}(z) = \mathbf{Q}^{(I_{(\alpha)})} \mathbf{S}^{(I_{(\alpha)})''}(z) \mathbf{Q}^{(I_{(\alpha)})H}$ is $\tau_{\max}^{(I_{(\alpha)})} - |\tau^{(I_{(\alpha)})}| = 0$, or when $\mathbf{S}^{(I_{(\alpha)})}(z)$ is sufficiently diagonalised with dominant off-diagonal column norm according to (8). Alternatively, the restricted update process ends if the total number of iterations of RU-SMD exceeds some user-defined value, I .

Fig. 1 demonstrates the restricted update step for $M = 5$, $\tau_{\max}^{(i)} = 3$. As can be seen, after three iterations, the maximum lag of the matrix in Fig. 1(i) is equal to zero; thus, $I_{(\alpha)} = 3$.

Note that $\mathbf{S}^{(i)}(z)$ will typically have fewer lags than the equivalent matrix in the i th iteration of the traditional SMD algorithm; thus, the search to identify the $k^{(i)}$ th column and row in the proposed approach may produce an inferior result to the search in SMD. However, we demonstrate that this does not significantly affect algorithm convergence in Sec. IV-C.

C. Matrix Regeneration

For index α , in the $I_{(\alpha)}$ th iteration of RU-SMD, following a restricted update step, a matrix $\mathbf{R}_{(\alpha+1)}(z)$, which is more diagonal than $\mathbf{R}_{(\alpha)}(z)$, is generated according to

$$\mathbf{R}_{(\alpha+1)}(z) = \mathbf{F}_{(\alpha)}(z) \mathbf{R}_{(\alpha)}(z) \tilde{\mathbf{F}}_{(\alpha)}(z), \quad (13)$$

where $\mathbf{F}_{(\alpha)}(z)$ is the concatenation of the elementary paraunitary matrices generated in the α th instance of *Restricted_update*. A matrix $\mathbf{F}'_{(\alpha+1)}(z)$ is also updated, which is a concatenation of the paraunitary matrices generated for indices $0 \dots \alpha$ and initial matrix $\mathbf{Q}^{(0)}$:

$$\mathbf{F}'_{(\alpha+1)}(z) = \mathbf{F}_{(\alpha)}(z) \cdots \mathbf{F}_{(0)}(z) \mathbf{Q}^{(0)} = \left(\prod_{x=0}^{\alpha} \mathbf{F}_{(\alpha-x)}(z) \right) \mathbf{Q}^{(0)}.$$

If $i > I$, or (8) is satisfied, the RU-SMD algorithm ends with $\mathbf{D}(z) = \mathbf{R}_{(\alpha+1)}(z)$ and $\mathbf{F}(z) = \mathbf{F}'_{(\alpha)}(z)$.

D. Algorithm Complexity

At iteration i of *Restricted_update* within RU-SMD, the number of MACs required to generate $\mathbf{S}^{(i)}(z) =$

$\mathbf{Q}^{(i)} \mathbf{S}^{(i)'}(z) \mathbf{Q}^{(i)H}$ can be approximated by $2L\{\mathbf{S}^{(i)'}\}M^3$, where $L\{\mathbf{S}^{(i)'}\}$ is the length of $\mathbf{S}^{(i)'}(z)$. To update $\mathbf{H}^{(i)}(z)$, $L\{\mathbf{H}^{(i)'}\}M^3$ MACs are required. Note that $L\{\mathbf{H}^{(i)}\}$ is reset to one following matrix regeneration. The cumulative complexity of *Restricted_update* is therefore approximately

$$C_{\text{RU}}(I) = M^3 \sum_{i=1}^I (2L\{\mathbf{S}^{(i)'}\} + L\{\mathbf{H}^{(i)'}\}). \quad (14)$$

During matrix regeneration, $\mathbf{F}'_{(\alpha+1)}(z) = \mathbf{F}'_{(\alpha)}(z) \mathbf{F}'_{(\alpha)}(z)$ and $\mathbf{R}_{(\alpha+1)}(z) = \mathbf{F}'_{(\alpha)}(z) \mathbf{R}_{(\alpha)}(z) \mathbf{F}'_{(\alpha)}(z)$ are computed. The former requires approximately $(L\{\mathbf{F}_{(\alpha)}\} + L\{\mathbf{F}'_{(\alpha)}\} - 1)M^3$ MACs, and the latter requires approximately $(2L\{\mathbf{F}_{(\alpha)}\} + L\{\mathbf{R}_{(\alpha)}\} - 2)M^3$ MACs; thus, the cumulative complexity of matrix regeneration for β total regenerations in RU-SMD is approximately

$$C_{\text{MR}}(\beta) = M^3 \sum_{\alpha=0}^{\beta-1} (3L\{\mathbf{F}_{(\alpha)}\} + L\{\mathbf{F}'_{(\alpha)}\} + L\{\mathbf{R}_{(\alpha)}\} - 2).$$

The total cumulative complexity of RU-SMD can therefore be approximated as

$$C_{\text{RU-SMD}}(I, \beta) = C_{\text{RU}}(I) + C_{\text{MR}}(\beta). \quad (15)$$

If the savings made during the restricted update step are larger than the overheads added by the matrix regeneration step — i.e., if $(C_{\text{SMD}}(I) - C_{\text{RU}}(I)) > C_{\text{MR}}(\beta)$ — the total cumulative complexity of RU-SMD will be lower than SMD.

IV. RESULTS

To benchmark the proposed approach, this section first defines the performance metrics for evaluating the SMD and RU-SMD methods before setting out a simulation scenario, over which an ensemble of simulations will be performed.

A. Performance Metrics

Since SMD and RU-SMD iteratively minimise off-diagonal energy, a suitable metric $E_{\text{norm}}^{(i)}$, defined in [11], is used; this metric divides the off-diagonal energy in the parahermitian matrix at the i th iteration by the total energy. Computation of $E_{\text{norm}}^{(i)}$ generates squared quadratic covariance terms; therefore a logarithmic notation of $5 \log_{10} E_{\text{norm}}^{(i)}$ is employed.

Equations (12) and (15) allow cumulative complexity measurements to be made at each iteration of SMD and RU-SMD.

B. Simulation Scenario

The simulations below have been performed over an ensemble of 10^3 instantiations of $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$, $M \in \{10; 20\}$, based on the randomised source model in [11]. In this source model, the order of $\mathbf{D}(z)$ is 118 and the order of $\mathbf{F}(z)$ is 60, such that the total order of $\mathbf{R}(z)$ is 238. The dynamic range is constrained to ensure that the average is around 30 dB.

Each algorithm was executed for $I = 200$ iterations with a stopping threshold of $\epsilon = 10^{-6}$ and truncation parameter of $\mu = 10^{-6}$. At every iteration of both implementations, the diagonalisation and cumulative complexity metrics defined in Sec. IV-A were recorded alongside the elapsed execution time. The length of $\mathbf{F}(z)$ was recorded upon each algorithm's completion.

C. Diagonalisation

The ensemble-averaged diagonalisation was calculated for the standard SMD and proposed RU-SMD implementations.

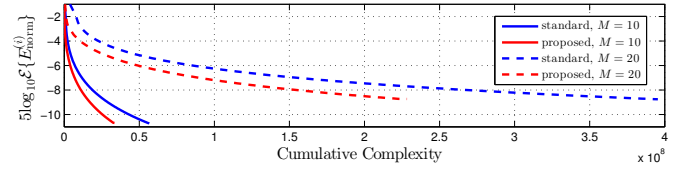


Fig. 2. Diagonalisation metric vs. cumulative algorithm complexity for the proposed and standard implementations for $M \in \{10; 20\}$.

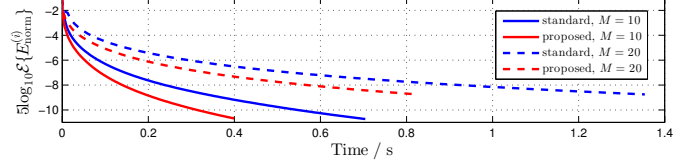


Fig. 3. Diagonalisation metric vs. algorithm execution time for the proposed and standard implementations for $M \in \{10; 20\}$.

The diagonalisation performance versus cumulative complexity and time for both methods are shown in Figs. 2 and 3, respectively. The curves of Fig. 2 demonstrate that for $M \in \{10; 20\}$, the proposed implementation operates with a lower cumulative complexity than the standard realisation, and is able to achieve a similar degree of diagonalisation — indicating that convergence is not affected by the use of a restricted update procedure. In addition, Fig. 3 shows that the lower complexity associated with the proposed approach translates to a faster diagonalisation than observed for SMD.

D. Paraunitary Filter Length

The ensemble-averaged paraunitary filter lengths were calculated for both algorithms. For $M = 10$, $\mathbf{F}(z)$ from SMD and RU-SMD was of length 84.4 and 87.1, respectively. Similarly for $M = 20$, lengths of 70.9 and 73.0 were observed for SMD and RU-SMD. The paraunitary filters generated by RU-SMD were therefore slightly longer than those from SMD.

V. CONCLUSION

In this paper, we have introduced a novel restricted update sequential matrix diagonalisation (RU-SMD) algorithm. This algorithm can produce the same accuracy of decomposition as SMD, but with decreased computational complexity. Simulation results underline that the same diagonalisation performance can be achieved by both methods, but within a shorter execution time for RU-SMD. While RU-SMD gives a slight increase in paraunitary filter length, it is not significant enough to negate the performance gains made elsewhere.

When designing PEVD implementations for real applications, the potential for the proposed RU-SMD algorithm to reduce time and complexity requirements offers benefits. In addition, the restricted update approach proposed here can be extended to any iterative PEVD algorithm in [8], [12]–[15] by adapting the update and matrix regeneration steps accordingly.

ACKNOWLEDGEMENT

Fraser Coutts is the recipient of a Caledonian Scholarship; we would like to thank the Carnegie Trust for their support.

This work was supported in parts by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.

REFERENCES

- [1] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.
- [2] C. H. Ta and S. Weiss. A design of precoding and equalisation for broadband MIMO systems. In *Asilomar SSC*, pp. 1616–1620, Pacific Grove, CA, USA, Nov. 2007.
- [3] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, 1993.
- [4] S. Weiss, S. Bendoukha, A. Alzin, F. Coutts, I. Proudler, and J. Chambers. MVDR broadband beamforming using polynomial matrix techniques. In *EUSIPCO*, pp. 839–843, Nice, France, Sep. 2015.
- [5] A. Alzin, F. Coutts, J. Corr, S. Weiss, I. K. Proudler, and J. A. Chambers. Adaptive broadband beamforming with arbitrary array geometry. In *IET/EURASIP ISP*, London, UK, Dec. 2015.
- [6] M. Alrmah, S. Weiss, and S. Lambbotharan. An extension of the music algorithm to broadband scenarios using polynomial eigenvalue decomposition. In *19th European Signal Processing Conference*, pp. 629–633, Barcelona, Spain, Aug. 2011.
- [7] S. Weiss, M. Alrmah, S. Lambbotharan, J. McWhirter, and M. Kaveh. Broadband angle of arrival estimation methods in a polynomial matrix decomposition framework. In *IEEE 5th Int. Workshop Comp. Advances in Multi-Sensor Adaptive Proc.*, St. Martin, pp. 109–112, Dec. 2013.
- [8] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE TSP*, 55(5):2158–2169, May 2007.
- [9] S. Icart, P. Comon. Some properties of Laurent polynomial matrices. In *IMA Int. Conf. Math. Signal Proc.*, Birmingham, UK, Dec. 2012.
- [10] P. Vaidyanathan. Theory of optimal orthonormal subband coders. *IEEE TSP*, 46(6):1528–1543, June 1998.
- [11] S. Redif, S. Weiss, and J. McWhirter. Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices. *IEEE TSP*, 63(1):81–89, Jan. 2015.
- [12] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss. Multiple shift second order sequential best rotation algorithm for polynomial matrix EVD. In *EUSIPCO*, pp. 844–848, Nice, France, Sep. 2015.
- [13] J. Corr, K. Thompson, S. Weiss, J. McWhirter, S. Redif, and I. Proudler. Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices. In *IEEE Workshop on Statistical Signal Processing*, pp. 312–315, Gold Coast, Australia, June 2014.
- [14] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, and I. K. Proudler. Causality-Constrained multiple shift sequential matrix diagonalisation for parahermitian matrices. In *EUSIPCO*, pp. 1277–1281, Lisbon, Portugal, Sep. 2014.
- [15] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, and J. G. McWhirter. Reduced search space multiple shift maximum element sequential matrix diagonalisation algorithm. In *ISP*, London, UK, Dec. 2015.
- [16] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Row-shift corrected truncation of paraunitary matrices for PEVD algorithms. In *EUSIPCO*, pp. 849–853, Nice, France, Sep. 2015.
- [17] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms. In *SSPD*, pp. 1–5, Edinburgh, UK, Sep. 2015.
- [18] J. Foster, J. G. McWhirter, and J. Chambers. Limiting the order of polynomial matrices within the SBR2 algorithm. In *IMA Int. Conf. Math. Signal Proc.*, Cirencester, UK, Dec. 2006.
- [19] C. H. Ta and S. Weiss. Shortening the order of paraunitary matrices in SBR2 algorithm. In *ICICSP*, pp. 1–5, Singapore, Dec. 2007.
- [20] F. Coutts, J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Memory and Complexity Reduction in Parahermitian Matrix Manipulations of PEVD Algorithms. In *EUSIPCO*, pp. 1633–1637, Budapest, Hungary, August 2016.
- [21] F. Coutts, J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Complexity and Search Space Reduction in Cyclic-by-Row PEVD Algorithms. In *Asilomar SSC*, Pacific Grove, CA, Nov. 2016.
- [22] F. Coutts, J. Corr, K. Thompson, and I. Proudler. Divide-and-Conquer Sequential Matrix Diagonalisation for Parahermitian Matrices. Submitted to *SSPD*, London, UK, Dec. 2017.
- [23] F. Coutts, K. Thompson, S. Weiss, and I. Proudler. Analysing the Performance of Divide-and-Conquer Sequential Matrix Diagonalisation for Large Broadband Sensor Arrays. Submitted to *SiPS*, Lorient, France, Oct. 2017.